EAI Endorsed Transactions Preprint Research Article/Editorial

Which Model Fits Best? Comparing Algorithm Performance on Diverse Datasets for Sentiment Analysis

Pham Vu Hien[1]

[1]Phenikaa University

Abstract

Sentiment analysis is an important task in natural language processing with wide applications such as understanding customer opinions, monitoring social media and business intelligence. This study aimed to evaluate and compare the performance of several machine learning models for sentiment classification on datasets from different domains. Specifically, we employed Naive Bayes, XGBoost, Random Forest, LSTM and CNN techniques to predict the polarity of movie reviews from the IMDB database and financial data. Both datasets were preprocessed and split into training, validation and test sets. The deep learning models LSTM and CNN significantly outperformed traditional algorithms on the IMDB reviews, with LSTM achieving the highest accuracy of 84%. Interestingly, the simple Naive Bayes method yielded the best results on the financial data. These findings provide insights into selecting appropriate techniques depending on the characteristics of the target domain and data. Overall, this research highlighted the effectiveness of different machine learning approaches for sentiment analysis and their strengths for certain problem types, which can guide future applications and model development.

Keywords: Analysis sentiment, machine learning, deep learning, IMDB

doi:xx.yyy/trans.journalid.V.n.i

1. Introduction

Sentiment analysis is a common text classification task that aims to determine whether a given text has a positive or negative sentiment. It has wide applications in understanding customer reviews, social media monitoring, and public opinion mining. In this study, we perform sentiment analysis on two popular datasets - IMDB movie reviews and Twitter tweets - using several machine learning classifiers to compare their performance.

The IMDB dataset contains 50,000 highly polarized movie reviews labeled as either positive or negative. The Twitter dataset consists of tweets labeled as either positive or negative based on their sentiment towards a particular topic. We preprocess both datasets by removing punctuation, stopwords, and lemmatizing words.

We apply various supervised machine learning algorithms for sentiment classification - Support Vector Machines (SVM), Naive Bayes, Random Forest, Long Short-Term Memory networks (LSTM) and Convolutional Neural Networks (CNN). SVM maps inputs to high-dimensional feature spaces and finds the optimal separating hyperplane. Naive Bayes makes strong independence assumptions between features. Random Forest constructs decision trees using bagging and feature randomness.

LSTM is a deep learning model well-suited for sequence data like text, with the ability to learn long-term dependencies. CNN uses convolution and pooling operations to extract high-level semantic features from text. We implement these deep learning models using an embedding layer to convert words to dense vectors followed by the core network architecture.

We evaluate and compare the performance of these classifiers on the IMDB and Twitter datasets in terms of accuracy, precision, recall and F1 score. The results will help understand which models are most effective for sentiment analysis of natural language texts from different domains and provide insights into applying machine learning to tackle sentiment classification problems.

*Corresponding author. Email: 20010739@st.phenikaa-uni.edu.vn

2. Related work

2.1. Naive Bayes

Naive Bayes classifiers have proven to be a practical and effective approach for sentiment analysis due to their ability to train quickly on large text corpora while still achieving reasonably high levels of classification accuracy. Based on the Bayes theorem and an assumption that features are independent of one another, Naive Bayes models represent texts as bags-of-words and calculate the probabilities of class membership based on word frequencies in labeled training data. While its independence assumption is naive, Naive Bayes performs surprisingly well by leveraging heuristic attribute independence estimates. It is able to effectively identify sentiment-carrying words and their orientations in order to determine overall text polarity. As Naive Bayes can learn well even from sparse feature vectors with limited training data, it serves as a suitable baseline technique for real-time sentiment applications. Common variants like Multinomial Naive Bayes, which considers word frequencies, or Bernoulli Naive Bayes, which uses binary word presence, are often used to classify tweets, reviews and other documents as positive or negative sentiment with reasonably good accuracylevels despite Naive Bayes' simplicity compared to more advanced algorithms.

Dey et al. [3] offers a powerful solution for analyzing customer sentiment in product reviews. This study compared two prominent machine learning approaches, Naive Bayes and Support Vector Machine (SVM), for sentiment analysis of Amazon product reviews. The models were evaluated using statistical measurements, and SVM emerged as the superior approach. SVM demonstrated higher accuracy, precision, recall, and F1-score in classifying the sentiment of customer reviews. These findings provide valuable guidance for businesses and researchers seeking to leverage machine learning for sentiment analysis tasks.Anmol Nayak et al. [7] performed Sentiment analysis on a standard Movie reviews Twitter feed dataset to investigate the three prominent supervised learning classifiers: Naïve Bayes, Support Vector Machine (SVM) and Random Forest, and hence determine the most accurate of them based on the results obtained for positive and negative polarity tweets.

2.2. XGBoost

XGBoost has proven highly effective for sentiment analysis due to its powerful gradient boosting framework and techniques to optimize model training. As an ensemble algorithm, it builds multiple decision trees sequentially while leveraging the predictive power of previous trees to correct mistakes. For sentiment classification tasks involving tweets, reviews and other texts, XGBoost takes feature representations like word embeddings, n-grams and sentiment lexical scores as input. It then utilizes regularization methods such as shrinkage, subsampling and bagging to minimize overfitting as trees are added in a stage-wise fashion. This helps XGBoost consistently outperform other algorithms by achieving exceptional accuracy, precision and recall levels in evaluating sentiment of large text corpora. Its built-in parallel processing also enables efficient training of large and complex models on datasets containing millions of samples. With the ability to handle both sparse and dense data, carefully tuned hyperparameters, and a robust implementation, XGBoost has emerged as a highly effective approach for tackling sentiment analysis challenges involving unstructured natural language texts from various domains.

K.Aifah et al. [1] an approach for sentiment analysis using Extreme Gradient Boosting (XGBoost), a powerful machine learning algorithm. The goal is to examine public sentiment towards telemedicine applications in Indonesia, particularly during the COVID-19 pandemic. The study analyzes reviews of Halodoc, the most popular telemedicine application in Indonesia. Despite the imbalanced nature of the data, XGBoost achieved high accuracy (96.24%) without the need for specific techniques to handle imbalanced data. The findings provide insights into the public's perception of telemedicine applications and identify key areas for improvement. The study also employs a fishbone diagram to analyze the most common factors of user complaints. A.Samih et al. [8] introduced a novel method called Improved Words Vector for Sentiment Analysis (IWVS) to enhance the accuracy of sentiment classification. IWVS utilizes word embeddings to construct sentiment vectors, which are then fed into an XGBoost classifier. The proposed method outperforms baseline models, such as TF-IDF and Doc2vec, in terms of F1-score for sentiment classification. The combination of IWVS and XGBoost achieves the best performance among the evaluated models. The findings demonstrate the effectiveness of IWVS in capturing sentiment information and the suitability of XGBoost for sentiment classification tasks.

2.3. Random Forest

Random Forests have proven to be a highly effective machine learning approach for automated sentiment classification due to their ability to capture complex non-linear relationships in text data without reliance on extensive feature engineering. As an ensemble classifier, Random Forests build numerous decision trees during training, each made on a random subset of the samples and features. This introduces randomness that helps reduce variance and prevents overfitting, resulting in a more robust model. For sentiment analysis tasks on documents such as reviews, tweets, and forum posts, Random Forest intake features representing unigrams, syntax patterns, sentiment scores and more to classify polarity at both the document and aspect level. Their implicit feature selection in choosing the most important signals at each tree split allows prioritizing the most relevant linguistic cues for sentiment. With hyperparameters optimized, the averaging of predictions across many decision trees gives Random Forest classifiers a competitive edge in accuracy for benchmark sentiment analysis datasets.

G.Khanvilkar et al. [4] used sentiment analysis to analyze natural language and identify emotions expressed by humans. It is commonly formulated as a two-class classification problem, categorizing sentiments as positive or negative. However, ordinal classification provides a more nuanced approach by assigning sentiments to multiple levels, such as very negative, negative, neutral, positive, and very positive. This study proposes a system that determines the polarity of user reviews using ordinal classification. The system employs machine learning algorithms, specifically SVM and Random Forest, to classify sentiments. The resulting polarity is then utilized to provide personalized recommendations to users. Stephenie et al. [10] proposes a method for sentiment analysis of Tokopedia product reviews using the Random Forest method, achieving an accuracy of 97.38%. This method can be used by businesses to improve their products and services and by consumers to make more informed purchasing decisions.

2.4. LSTM

Sentiment analysis of social media posts, news articles, and other texts pertaining to LGBT topics can provide valuable insights into societal attitudes toward gender and sexual minorities over time. However, accurately classifying sentiment on LGBT issues poses unique challenges compared to general domains, as models must learn to identify nuanced sentiments and correctly interpret ambiguous or coded language. Training data needs to represent the full spectrum of sentiment, from strongly supportive to neutral to overtly intolerant, so classifiers are calibrated to the subtleties of LGBT discourse. Fine-tuning is also important to recognize sentiment-bearing emojis, terms, and expressions in LGBT contexts. While flagging explicitly derogatory or harassing language is a priority, models should avoid defining any view as inherently negative and focus only on statistical patterns from carefully labeled data. Addressing these issues can help make online spaces more positive while also enabling monitoring of societal progress on LGBT acceptance through sentiment trends at both the document and topic levels. With appropriate safeguards, LGBT sentiment analysis shows promise for gaining a deeper understanding of public perceptions on important issues of inclusion, rights and equality over time.

RK Behera et al. [2] proposed a hybrid approach of two deep learning architectures namely CNN and LSTM (RNN with memory) for sentiment classification of reviews posted at diverse domains. It is highly adaptable in examining big social data, keeping scalability in mind, and secondly, unlike the conventional machine learning approaches, it is free from any particular domain. The proposed method achived an accuray of 97.04% when analyzing social media data.Murthy et al. [6] proposes a sentiment analysis model using LSTM (Long Short-Term Memory) to analyze customer reviews on Tokopedia, a popular e-commerce platform in Indonesia. The model achieves an accuracy of 97.38% and outperforms other machine learning methods such as SVM, Naive Bayes, and Logistic Regression.

2.5. CNN

Convolutional neural networks have become a highly effective approach for sentiment classification due to their ability to automatically learn relevant linguistic features from text data. In CNN sentiment classification models, words are first embedded as vectors then passed through a series of convolutional layers to extract important n-gram patterns. Max pooling layers help reduce overfitting by summarizing these features before sending to fully connected layers for final classification. By handling variable length inputs and capturing local semantic and syntactic contexts, CNNs can learn robust representations of sentiment from large amounts of unlabeled data in an end-to-end fashion without necessity of manual feature engineering. This has allowed CNN sentiment classifiers to achieve state-of-the-art results on numerous benchmark datasets involving reviews, tweets and other documents. Variations continue to be explored through combinations with recurrent layers to better capture long-range dependencies. With hyperparameters like filter sizes, number of filters and epochs carefully tuned for a given task and text domain, CNNs represent a highly accurate deep learning approach for automated sentiment analysis of unstructured natural language texts.

Vo, Quan-Hoang et al. [9] proposed a multi-channel LSTM-CNN model for Vietnamese sentiment analysis, which employs LSTM and CNN to generate information channels for capturing both local and global dependencies in a sentence. The proposed model outperforms SVM, LSTM, and CNN on two Vietnamese datasets, VS and VLSP, demonstrating the effectiveness of the multi-channel approach in enhancing sentiment classification performance. The paper also introduces a Vietnamese sentiment corpus (VS) containing 17,500 reviews/comments collected from Vietnamese commercial websites and annotated by three human annotators.. Mulyo et al. [5] proposed a Convolutional Neural Network (CNN)-based approach for aspect-based sentiment analysis, which can extract multiple aspects and their corresponding sentiments from a single sentence. The proposed approach outperforms several classical machine learning methods and achieves state-of-the-art results on two benchmark datasets. The proposed CNN-based approach uses a threshold to select the best data in training data, which enables the model to generate more than one aspect using one data test. The model also achieves a better F-Measure compared to CNN and three classic Machine Learning methods, namely SVM, Naive Bayes, and KNN. The proposed approach is effective for aspect-based sentiment analysis and can be applied to other natural language processing tasks.

3. Proposed Methods

3.1. Dataset

IMDB Dataset. The IMDB dataset is one of the most widely used benchmarks for evaluating sentiment classification models. It contains hundreds of thousands of movie reviews sourced from the Internet Movie Database website. The reviews are split into train and test sets, each containing an equal number of reviews labeled as either positive or negative sentiment. This creates a balanced binary classification problem for models to predict the overall sentiment polarity of reviews. The large dataset size allows deep learning algorithms to effectively learn features directly from text without extensive feature engineering. Models are assessed based on their ability to correctly classify new reviews not seen during training.

In addition to its widespread use for evaluating classifier performance, the IMDB dataset also provides insight into moviegoers' opinions at scale. The vast collection of reviews spanning many genres, eras and other metadata provides a lens into popular sentiment trends over time. For example, researchers have used the IMDB data to analyze how factors like a film's budget, ratings, genre or actors correlate with review positivity. The anonymity of the reviews also lends itself to studying sentiment free from biases of professional critics. As a user-generated opinion dataset, the IMDB movie reviews remain a stap benchmark for the natural language processing task of sentiment analysis.

Financial Sentiment Analysis. The data is available on the Kaggle and intended for advancing financial sentiment analysis research. It's two datasets (FiQA, Financial

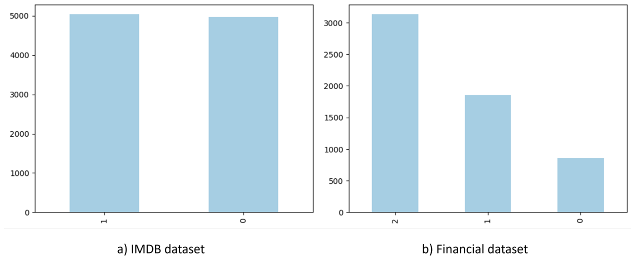

Figure 1. Number of every label of two datasets

PhraseBank) combined into one easy-to-use CSV file. It provides financial sentences with sentiment labels.

3.2. Models

Naive Bayes. The Naive Bayes algorithm is a supervised machine learning algorithm based on the Bayes' theorem. It is a probabilistic classifier that is often used in NLP tasks like sentiment analysis (identifying a text corpus' emotional or sentimental tone or opinion). The Bayes' theorem is used to determine the probability of a hypothesis when prior knowledge is available. It depends on conditional probabilities. The formula is given below:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where P(A|B) is posterior probability i.e. the probability of a hypothesis A given the event B occurs. P(B|A) is likelihood probability i.e. the probability of the evidence given that hypothesis A is true. P(A) is prior probability i.e. the probability of the hypothesis before observing the evidence and P(B) is marginal probability i.e. the probability of the evidence. When the Bayes' theorem is applied to classify text documents, the class c of a particular document d is given by :

$$c_{MAP} = argmax P(x_1, x_2, ..., x_n|c)P(c) \tag{1}$$

Let the feature conditional probabilities P(x_i | c) be independent of each other (conditional independence assumption). So,

$$P(x_1, x_2, ..., x_n|c) = P(x_1|c) \times P(x_2|c) \times ... \times P(x_n|c) \tag{2}$$

Now, if we consider words as the features of the document, the individual feature conditional probabilities can be calculated using the following formula :

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}} \tag{3}$$

$$\hat{P}(\mathrm{w}_i|c_j) = \frac{count(\mathrm{w}_i, c_j)}{\sum_{\mathrm{w} \in V} count(\mathrm{w}, c_j)} \tag{4}$$

But what if a given word w_i does not occur in any training document of class c_j, but it appears in a text document? P(w_i | c_j) will become 0, which means the probability of the test document belonging to class c_j will become 0. To avoid this, Laplace smoothing is introduced and the conditional feature probabilities are actually calculated in the following way :

$$\hat{P}(\mathrm{w}_i|c) = \frac{count(\mathrm{w}_i, c) + 1}{\sum_{\mathrm{w}\in V} count(\mathrm{w}, c) + |V|} \tag{5}$$

where |V| is the number of unique words in the text corpus. This way we can easily deal with unseen test words.

XGBoost. XGBoost is known for the ability to optimize the consumption of time, memory resources, and handle imbalanced data. The XGBoost or Extreme Gradient Boosting algorithm is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework. Ensemble learning offers a solution to combine the predictive power of multiple learners. In boosting, trees are built sequentially so that each next tree aims to reduce errors from the previous tree. Each tree learns from its predecessors and updates residual errors. Therefore, the tree that grows next in the sequence learns from the updated residuals. The base learners in boosting are weak learners in which the bias is high. Each of these weak learners contributes to give some information for prediction, enabling the boosting technique to produce a strong learner by effectively combining these weak learners. Suppose we have a training data x_i and their labels y_i, XGBoost utilize classifier to predict the final prediction $\hat{y}_i^t$.

$$\hat{y}_i^t = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{t-1} + f_t(x_i) \tag{6}$$

Where $\hat{y}_i^{t-1}$ is previous prediction and $f_t(x_i)$ is new prediction. To get a good model, in XGBoost we need to minimize the following objective function.

$$L^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega(f_t) \tag{7}$$

The objective function contains loss function $l(y_i, \hat{y}_i)$ and regularization term $\Omega(f_t)$. With the existing of Equation (6), we can rewrite the objective function as follow.

$$L^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) \tag{8}$$

The loss function measures how well the model fits on the training data, while regularization measures the complexity of trees. Optimizing loss function encourages predictive models for higher accuracy while optimizing regularization encourages generalized simpler models. Regularization is also utilized to avoid the model from overfitting.

Random Forest. Random Forest Algorithm is the advancement of Classification and Regression Tree (CART) method with the implementation of bootstrap aggregating (bagging) and random feature selection. Procedure of random forest algorithm on the data of n observations and p predictor. Random samples of size n are drawn with the possibility of obtaining the same data (with replacement). This phase is called bootstrap. Using the bootstrap samples, the tree is grown until the maximum size is reached, which is done without pruning. At each node, the random feature selection is used to determine the split, which m number of variables randomly sampled as candidates at each split must be m « p, at which point, the best node will be chosen based on m number of variables available for splitting.In order to determine the split used as root node/node, Gini Index is used in Random Forest method. The formula of Gini index can be described as following:

$$Gini = 1 - \sum_{i=1}^{k} p_i^2 \tag{9}$$

where p_i is probability of an attribute being classified to class i, k is total number of attributes being classified to a particular class.

LSTM. When considering the IMDB dataset, the presented approach passes through the three layers, such as word embedding, Convolution, and LSTM layer. In the first layer, word-embedding is applied to embed the words in the review, which eradicates the domain dependency of the data features. The second phase uses the Convolution layer and the pooling process in order to identify the important local and deep features in the sentence. The third layer applies the LSTM network on the output obtained from the second layer to capture their sequential dependency from left to right. The combination of three layers helps in realizing the behavior of the sentence. The output of 64 the LSTM is then supplied to the fully connected sigmoid layer to evaluate the result by considering binary cross-entropy as the loss function.

The order of the layers when applying the LSTM model to the seconda data - financial data is similar, but with the addition of 2 convolutional layers, 1 LSTM layer, and 2 layers after the LSTM before passing through the softmax activation function. The convolutional layers help to extract local features from the data, while the LSTM layer helps to learn the long-term dependencies in the data. The fully connected layers help to combine the features learned by the convolutional and LSTM layers and produce the final output of the model. The softmax activation function is used to convert the output of the fully connected layers into a probability distribution over the possible output classes.

CNN. The CNN model when applied to IMDB data and financial data consists of 1 embedding layer, 4 Conv1D layers, a flatten layer, then 2 Dense layers, and finally a classification layer with 3 units and a softmax activation function. The embedding layer converts the input text data into a dense vector representation. This allows the model to learn the meaning of words and phrases in the context of sentiment analysis. The Conv1D layers help to extract local features from the data. These features can include things like the presence of certain words or phrases, the order of words in a sentence, and the grammatical structure of a sentence. The flatten layer converts the 2D output of the Conv1D layers into a 1D vector. This allows the model to connect the features learned by the Conv1D layers to the output layer. The Dense layers help to combine the features learned by the Conv1D layers and produce the final output of the model. The first Dense layer typically has a large number of units, while the second Dense layer typically has a small number of units, equal to the number of output classes. The classification layer uses a softmax activation function to convert the output of the Dense layers into a probability distribution over the possible output classes. In the case of sentiment analysis, the output classes are typically positive, negative, and neutral.

Method	Accuracy	Recall	F1 Score
Naive Bayes	63.5 %	63.9 %	63.061 %
XGBoost	83.45 %	83.395 %	83.419 %
Random Forest	84 %	83.932 %	83.963 %
LSTM	**84.100 %**	**84.129 %**	**84.095 %**
CNN	81.7 %	81.644 %	81.661 %

Table 1. Accuracy, Recall and F1 Score of IMDB dataset

Method	Accuracy	Recall	F1 Score
Naive Bayes	66.895 %	**59.44 %**	**59.32 %**
XGBoost	**69.8 %**	56.484 %	57.948 %
Random Forest	62.19 %	40.373 %	36.812 %
LSTM	62.7 %	57.404 %	56.537 %
CNN	61.42 %	58.243 %	56.22 %

Table 2. Accuracy, Recall and F1 Score of financial dataset

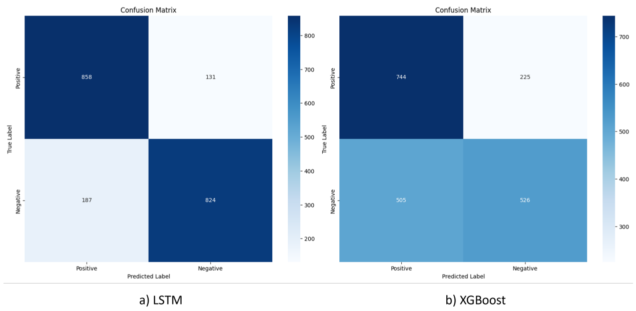

Figure 2. Cofussion matrix between LSTM model and XGBoost with IMDB data

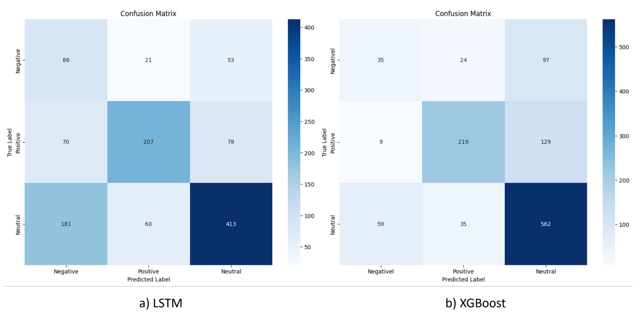

Figure 3. Cofussion matrix between LSTM model and XGBoost with financial data

4. Results

Two datasets were divided into training, validation and test sets in a 60:20:20 split ratio. Through training different models on the data, I found that the LSTM model was able to achieve the highest accuracy after only 30 epochs of training, while the CNN model required 100 epochs to reach its peak performance.

When evaluating the performance of the different models on specific datasets, the LSTM architecture worked best for sentiment classification on the popular IMDB movie review dataset, attaining an impressive 84% accuracy. Meanwhile, the simpler Naive Bayes classifier yielded the highest results on a financial/economic dataset, suggesting it may be better suited for that type of classification problem involving financial or numeric predictions.

In summary, through experimenting with different deep learning and traditional models on split portions of my full dataset, I was able to determine that LSTM outperformed other methods like CNN and Naive Bayes for sentiment analysis, but a statistical approach worked best for financial classification. The number of training epochs also varied by model, with LSTM needing fewer iterations to optimize compared to CNN.

5. Conclusion

In this study, we evaluated various machine learning classifiers for sentiment analysis of text data from different domains. Specifically, we applied Naive Bayes, XGBoost, Random Forest, LSTM and CNN models to perform sentiment classification on movie reviews from the IMDB dataset and financial data. Our experimental results showed that deep learning models generally outperformed traditional algorithms. The LSTM architecture achieved the highest accuracy of 84% on the IMDB reviews, demonstrating its effectiveness in capturing long-term dependencies important for sentiment analysis of natural language text. Meanwhile, Naive Bayes yielded the best performance on financial data, indicating statistical models may be more suitable for classification problems involving numeric data. These findings provide insights into selecting optimal machine learning approaches depending on the characteristics of the target domain and data. Overall, this research highlighted the importance of comparing different algorithms to identify the most accurate model for a given sentiment analysis task. Future work includes exploring ensemble and hybrid methods to potentially further improve classification performance.

References

[1] Khansa Afifah, Intan Nurma Yulita, and Indra Sarathan. "Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier". In: *2021 International Conference on Artificial Intelligence and Big Data Analytics*. IEEE. 2021, pp. 22–27.

[2] Ranjan Kumar Behera et al. "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data". In: *Information Processing & Management* 58.1 (2021), p. 102435.

[3] Sanjay Dey et al. "A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews". In: *2020 International Conference on Contemporary Computing and Applications (IC3A)*. 2020, pp. 217–220. DOI: 10.1109/IC3A48958.2020.233300.

[4] Gayatri Khanvilkar and Deepali Vora. "Sentiment analysis for product recommendation using random forest". In: *International Journal of Engineering & Technology* 7.3 (2018), pp. 87–89.

[5] Budi M Mulyo and Dwi H Widyantoro. "Aspect-based sentiment analysis approach with CNN". In: *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE. 2018, pp. 142–147.

[6] GSN Murthy et al. "Text based sentiment analysis using LSTM". In: *Int. J. Eng. Res. Tech. Res* 9.05 (2020).

[7] Anmol Nayak and D Natarajan. "Comparative study of naive Bayes, support vector machine and random forest classifiers in sentiment analysis of twitter feeds". In: *International Journal of Advance Studies in Computer Science and Engineering (IJASCSE)* 5.1 (2016), p. 16.

[8] Amina Samih, Abderrahim Ghadi, and Abdelhadi Fennan. "Enhanced sentiment analysis based on improved word embeddings and XGboost." In: *International Journal of Electrical & Computer Engineering (2088-8708)* 13.2 (2023).

[9] Quan-Hoang Vo et al. "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis". In: *2017 9th international conference on knowledge and systems engineering (KSE)*. IEEE. 2017, pp. 24–29.

[10] Budi Warsito, Alan Prahutama, et al. "Sentiment analysis on Tokopedia product online reviews using random forest method". In: *E3S Web of Conferences*. Vol. 202. EDP Sciences. 2020, p. 16006.